$2.00

THE JACG NEWSLETTER

THE JERSEY ATARI COMPUTER GROUP

VOLUME 8 NUMBER 12 BBS : 201-298-0161 FEBRUARY 1989

FROM THE EDITOR'S DESK

I have had considerable opportunity to read numerous newsletters from other ATARI user groups. It has never ceased to amaze me as to how our JACG has been able to so easily integrate both 8-Bit and 16-Bit users into one homogeneous group. It appears that many groups have either split into two; or (even worse), dropped coverage of one or the other computer. One thing that I can't understand...is why, in the light of continuing ST sales here in the U.S., there are not more 16-Bitters joining our group. Even more strange, I now get calls for help from "new" 8-Bitters who are not members of any user group. Of course my help is not contingent upon their becoming a member of the JACG...but I do let them know of our existence!

It appears that my plea for someone to take over the editorial helm has been heard, and that someone with prior ATARI user group newsletter editorial experience will take over in a couple of months...I'll keep the readership informed of developments in that area. When that in fact becomes a fact, I will move into the advertising area and try to stimulate new advertising in our Newsletter.

Just a reminder, anyone desirous of making demonstrations at our monthly meetings should set up same with the appropriate vice president (8 or 16-Bit). This will ensure preservation of a time slot, and allow for better planning of meeting format. The vice presidents thank you in advance!

In the kudo department...let's not forget OUR disk librarian...Sam Cory. Although unable to make the monthly meetings, the time and effort that he spends to further the success of the JACG is legion. Thank you, Sam, for your continuing support!

IN THIS ISSUE ...

CALENDAR OF EVENTS

NEXT MEETING

MARCH 11, 1989

GEMINI

PROUDLY ANNOUNCES ITS APPOINTMENT AS AN ATARI BUSINESS COMPUTER CENTER

THE ONLY OFFICAL ATARI MEGA & 1040ST DEALER IN NORTHERN NJ

JACG SPECIAL

ATARI SX212 MODEM

300/1200 BAUD
FULL HAYES COMPATIBLE
XL/XE, ST, MEGA & IBM

$59.95

	MONO SYSTEM	COLOR SYSTEM	CPU ONLY
520 STFM w/DS DRIVE	699.95	849.95	599.95
1040 ST w/DS DRIVE	849.95	999.95	OVER 850 "ST" SOFTWARE TITLES IN STOCK
MEGA 2ST	1249.95	1399.95	
MEGA 4ST	1749.95	1899.95	

WE STILL CARRY A COMPLETE SELECTION OF XL/XE HARDWARE & OVER 1000 SOFTWARE TITLES!

GEMINI ENTERPRISES

NEW LOCATION:
171 RIDGEDALE AVE
MORRISTOWN, NJ

267-098

TOLL-FREE ORDERS ONLY (800)543-381

PRESIDENT'S REPORT
by Gary Gorski

PRESIDENTS REPORT
FEB 1989

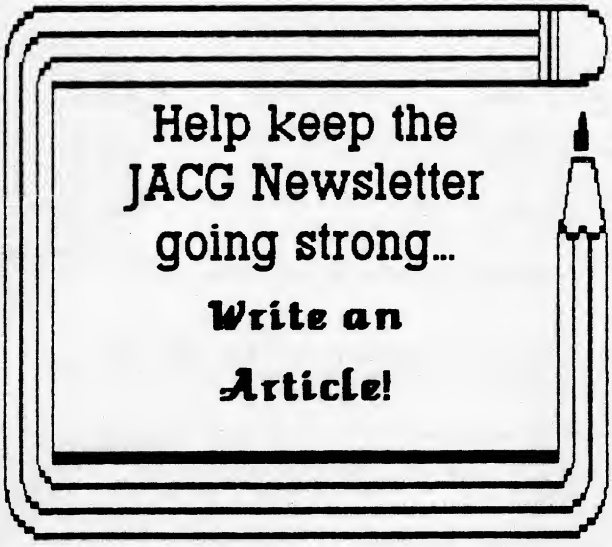
This month has been an exceptionally busy one for me. Alot of things are in the works for the upcomming JACG meetings. All of this takes time, planning,and organization. I would like to thank the Executive Board for all the time, trouble, and sweat that each Board member has put into the smooth running of the JACG organization. Just some of the things that are planned for the upcoming meetings are:

1. Telecommunications (March Meeting
2. Information
3. MIDI
4. Oddities (More on these next month...)

Remember, it's never to early to sign up to do a demo at one of our meetings. You can contact, Dave, John, or myself at a meeting, or by phone, and of course we can be reached on the JACG BBS: (201) 298-0161.

Last month Dave Noyes worked alittle overtime to bring us a 20 page Newsletter, (Thanks Dave !). In order to continue with AT LEAST A 20 PAGE NEWSLETTER we (JACG) will be inserting something new, starting with the March Newsletter. Z-Net will become part of our monthly issue. Z-Net will be the hard copy issue from Ron Kovacs, editor of Zmagazine and STZmagazine, which by the way are available on GEnie and the JACG BBS. Z-Net will include, news, reviews, and articles from both the Atari 8 and 16 bit community. Welcome aboard Z-Net !

And lastly, please fill out the questionairre that Dave has supplied to all of us about ANALOG. We will be sending them back as a group. I would like to thank all the JACG that have donated door prizes for the monthly meetings, they are always welcome! Well, that about wraps things up for the month in review. See everyone at the meeting, until then, happy computing!



NOISE FROM NOYES
by Dave Noyes

Well, to be honest, it is sometimes difficult to sit down and be creative, especially on cue! And, as another month has rolled around (and do they ever roll around!), I try to focus on a little more Noise from Noyes. I don't view the column as "filler" (ugh!), and if I had nothing to say, I would say nothing. However, hardly ever being at a loss for words...here comes some NOISE.

I picked up the SPARTA DOS-X cartridge at GEMINI the other day, for $59.95...which I consider a bargain (lists at $79.95). I've only been able to play with the cartridge for the past weekend, and will be unable to provide an appropriate review, other than to say that it is well worth the investment...look for more on it from me next month. I got a chance to see ATARI DOS XE...I'll let Neil Van Oost's review in this issue address it. Suffice it to say that DOS XE will not be my DOS of choice!

Another recent purchase from GEMINI ($59.95) was the ANIMATION STATION from SUNCOM. Yeah, I know, I already had the KOALA PAD, and some of you might say that the purchase of the STATION was redundant. Then why buy it? Well, for one thing, the STATION has a larger work area than the PAD. Further, the software provided with it is actually BAUDVILLE's BLAZING PADDLES...configured to be paddleless, accepting input only from the STATION.

I have also recently purchased the ATARI XEP-80 80 column board, and the 80 column word processor,TURBOWORD, by MICROMISER. You know, the one that Shree Vandenberg demo'd a few months ago. I (yes I) reviewd this in the Mar/Apr ATARI EXPLORER. If you don't subscribe, please buy a copy...or better, subscribe! TURBOWORD is reason enough to purchase the XEP-80! A bonus...on COMPUSERVE I found ATERM880.COM, an 80 column terminal program for the XEP-80. I now go online in 80 columns!

.....'til next month....

Sightseeing with

FLIGHT SIMULATOR II
#11

continued:

Dave Arlington - JACG

When you get there and get ready to make the turn, be sure to check on radar and not turn too soon. It's easy to misjudge these guys if you get too close and end up a smudge. As you turn around the mountain, be sure to take plenty of views out your side windows so you get some idea of how long and big this mountain is. Pretty neat, huh? I thought this made up for no Lock Haven.

As you fly along the north side of the mountain, tune in your NAV radio to the Phillipsburgh (Pennsylvania, not New Jersey) VOR at 115.5. (See last month's article for instructions.) Use Control-V and (<, >) signs to get the little vertical line centered with the radio reading TO. As you fly along, you'll notice the needle or line keeps moving away from the center and you'll have to keep readjusting to keep it centered. Don't change course yet though. The needle keeps moving from center becuase we are flying almost perpendicular to the VOR station, rather TO it or FROM it.

As you pass the western edge of the mountain, get the line centered again and turn to the heading indicated on the top of the NAV radio. If you look on the Detroit area chart that comes with the disk, you'll see the airport we're going to be landing at, Mid-State.

Your distance measuring equipment should show you to be some 20+ miles out from the VOR tower. As you get closer on your heading, you'll notice we seem to be heading straight for that mountain that appeared ahead! Hmmm.. This seems tricky. No sign of the airport yet. Maybe it's on the other side of the mountain. If so, that poses two problems. First we have to get over the mountain. Second, if it's on the other side, we'll have to get down to the ground in a awful hurry on the other side.

I can help with the first problem. FRom my vast store of knowledge (and my brother-in-law's real-life sectional map) I happen to know that mountain is 2,749 feet high, cutting it pretty close at our present altitude. So put on a notch or two of power and climb to 3000-3100 feet. Take that notch or two off again when you reach that hieght. Don't get much higher since Mid-State airport is at 1948 feet altitude.

About 10 miles out, you can spot the airport off to the right and heave a sigh of relief. You don't have to go over the mountain after all. The airport is nestled almost right at the bottom of two mountains, the west edge of the one we're flying towards and north of another one that just popped up. We'll land on runway 16 (heading 160) so immediately turn right to a heading of 250 degrees to approach it at a 90 degree angle. This is what is called the 'base' leg of a standard airport pattern, technical talk, I know.

As soon as you get on the heading of 250, slow the plane down to 80 knots by dropping power 7 notches and slowly giving 4 presses of up elevator. Keep you eye on the runways out the left front window. When the runway is about 1/3 of the way across your direct left side window, turn to the heading of 160 degrees. If all goes well, you should be lined up perfectly for it. Landing is accomplished by following the step by step instructions from my previous article.

When you get safely(!) on the ground, look out the front window at that mountain and think about trying to land on this runway from the opposite direction. You would have to fly over the mountain in that case and probably wouldn't see the runway until the last second.

Now you hopefully will admit that even though we did not see the birth place of our planes this fine morning, all that mountain and river flying was worth getting up so early. See you in the skies soon!

JERSEY ATARI COMPUTER GROUP TREASURER'S REPORT
FOR THE PERIOD JANUARY 1 TO DECEMBER 31, 1988

* Revised balance as of December 31, 1987			$2,163.
INCOME:			
Membership			
3rd class	$3,096.00		
1st class	$1,069.00	$4,165.00	
Library Sales			
8 bit	$1,502.00		
16 bit	$1,307.00	$2,809.00	
Advertising		$79.00	
Newletter Sales		$14.00	$7,067
EXPENSES:			
Postage		$1,017.18	
Newsletter		$4,589.23	
Disks for Library		$1,936.00	
Programs		$123.75	
Communications		$40.00	
Awards and Prizes		$62.00	
Misc. Supplies		$444.55	
Computer Show		$166.00	
Computer Repair		$270.30	
Bank Charges		$37.90	
Sales Tax		$116.18	$8,803.
BALANCE IN NATIONAL COMMUNITY BANK 12/31/88			$427.

* Actual balance in Crestmont Fed. Sav. Bnk. was not established until the account was closed in May 1988.

Respectfully submitted

Jack S. Rutt
Treasurer

THE MAGIC OF COMPOUND INTEREST

BY ERIC JACOVES

Have you ever thought about your financial future?? When ira's were possible did you take advantage of them or didyou say to yourself that you couldnt spare the $2000.00 each year for the five years that the government allowed you to take $2000 and remove it from your taxes and avoid paying taxes on the interest untll you were retlred and in a much lower tax braket.

The program in thls Newsletter is based on the mathmatical formula to calculate the future value of an investment. The formula is...

A = P*(1-R/Q)^N*Q

Where A = The future amount
P = the principal invested
R = the rate of return(interest)
Q = how many tlmes the interest is compounded each year
N = the number of years for the investment

or A = P(times)(1-R/Q)to the N*Q power
The program is wrltten In GFA basic but it can readlly be converted lnto standard st basic and Atari 800 basic as well. The only statements that need be changed are the lprint using statements to go to Atari 800 basic.

The program has two optlons. the first is to simply calculate the future value of any current sum at the glven interest rate and with the number of compoundings per year. It does not include taxes that would have to be paid each year on the interest earned, This option is actually option 2 in the program. Option 1 allows you to see how your money grows over the years and you can give the program a tax rate and the program will automatically calculate your taxes on the interest and deduct those nasty taxes from the principal for next year. to see the effect of taxatlon just run the program once wlth your tax bracket and again wlth a zero tax bracket. The program produces four columns of output, the year, the new principal, what the current principal grows to over the year, and lastly the taxes that are removed from the new prlncipal. You should enter the tax rate and the interest rate as whole numbers since the program devldes your numbers by 100 to convert them to the proper decimal form for the calculations. An interest rate under 10% should be entered as 5.25 for .0525 real declmal rate.

The example shows what would have happened lf a 20 year old was smart enough to take advantage of the IRA's when he could or can lf he has no pensIon at work. He puts $2000.00 In per year and I assumed that he could get 12% interest(not at the bank of course) as you can see from the table he wlll have $3,810,193 at age 65 wlch is a very nice way to retlre wlth absolutly no worrles. Guess how many Amerlcans are smart enough to make good on this. Even lf you have to pay taxes its still a very good idea to start a savings plan when you are young for as table two shows, even wlth a 25% tax rate you could stlll amass $1,387,812 with a mere $2000 per year and that is not small potatoes either. The fact that the government took away $2,422,381 from you Is enough to drive you to drlnk or worse but that is the way it is for us.

You can also track an investment that stops adding principal after a few years for you tell the program the number of years that you will be adding funds to the investment. Have fun with It and lf you are not able to convert the program to your machine let me know and I wlll provide you with a runable versIon for any Atarl computer....

Eric Jacoves

```
Rem a program to compute the future value of an investment
Rem By Eric Jacoves
Print "The Principal is ??."
Input P
Print "How Much Will You Add Next Year ?"
Input Ayr
Print Using "  for how many years will you add #### ",Ayr," per year ?"
Input Nyrs
Print "The Rate of Interest is ? "
Input R
R=R/100
Print "The number of times compounded per year is "
Input Q
Print "how many years"
Input N
Print "now the rip off , what tax braket are you in ? "
Input Tax
Tax=Tax/100
Print "do you want to see how the money grows ? 1 = yes, 2 = no, which?"
Input Ny
If Ny=1 Then
  Goto Grow
Endif
A=P*(1+R/Q)^(N*Q)
Lprint "The future value is ",A
Goto Nxt
Grow:
Print "how many years ??"
Input J
Lprint "Interest rate is ";R
Lprint "Compounded ",Q;"  times per year"
Lprint Using "Tax Rate is #.## ",Tax
Lprint Using "With #### Added Each Year For ## Years",Ayr,Nyrs
Lprint
Lprint "   Year         Principal         Grows to     Tax "
For I=1 To J
  A=P*(1+R/Q)^Q
  Intrest=A-P
  Rip=Intrest*Tax
  Lprint Using "   ##    ############.##    ############.##  ########.##",I,P,A
  P=A-Rip
  If Nyrs>0 Then
    If Nyrs>I Then
      P=P+Ayr
    Endif
  Endif
Next I
Lprint
Lprint
Nxt:
End
```

EXECUTIVE MEETING MINUTES

February 3, 1989
J.A.C.G. SECRETARY MICHAEL D. Hochman

The meeting was called to order by the President at 8:00PM with Gary Gorski(President), Dave Noyes(VP 8-bit and Newsletter Editor), Sam Cory(Librarian) and Mike Hochman(Secretary and Membership Chairman) present and constituting a quorum.

Sam Cory began the meeting by suggesting member reviews and articles on XEGS (the ATARI XE Game System). As many of us know, XEGS is actually an ATARI XE computer disguised as a game machine.

In light of a desire on ATARI's part to better relations with User Groups, they have expressed a willingness to provide speakers for meetings, we will be trying to get Sig Hartmann of Atari to visit in the future. Mr. Hartmann was at onetime the head of new software development at ATARI and is now an ambassador on the user group scene. One condition may be the necessity for a joint meeting to ensure a large turnout. This involves contacting other groups from neighboring areas and establishing a date and place for this event.

There will be a reassesment of the JACG Library disks for quality of content and integrity of data. This will ensure that JACG continues to produce PD disks of the best programs, and that these disks are fully functional.

The Secretary would like to know if someone would be interested in assisting with the Newsletter Exchange. That person would have to maintain the Newsletters we receive from other clubs. This involves alphabetizing, binding and distributing them at the meetings. A list must also be kept of members who have checked out a Volume of the Newsletter.

The committee has agreed on accepting a magazine insert to be included with our JACG Newsletter. "Z-NET" as it is called, will pay us at least 20 cents per printed newsletter to carry their insert which will include news, articles, reviews, and advertising. This will cover the cost of having the insert in our newsletter. Full details will be given at the meeting.

We have approved the purchase of 500 5-1/4" diskettes and a supply of mailers for the 8-bit Library.

The theme approach to meetings seems to be very well liked idea. The following are future theme meetings in the works:

1. ODDITIES- Members will be encouraged to display and/or demonstrate odd or rare contraptions used with their ATARIs. Please see Mike Hochman as the has volunteered to chair this meeting.
2. INFORMATION- ATARI personalities from the media and ATARI itself will be asked to speak and inform.
3. MIDI- A meeting totally devoted to Musical applications.
4. TELECOMMUNICATIONS- BBSs, Modems, Software, etc. which allow the ATARI to communicate with other computers.

The meeting was adjourned at 10:15PM.

FEBRUARY DOM

16-Bit

UTILITY DISK

JACG LIBRARY VOLUME # 47
John H. Dean - JACG

This disk was brought into our disk library by C. Meadows, and contains a bunch of utilities. Most of them have DOC files. The README.DOC file on the disk is as follows:

"As all files are ARC'ed, a short explanation of how to get them out of ARC. Double click on ARC.TTP to run the program. When the dialogue box appears,KEY ON caps lock and type XH Program_name.ARC *.*. This extracts any files in the ARC'ed file and prints them to the same disk as ARC.TTP. To change drives, add the extender B:\ before the wild cards. Un-ARC ARC.ARC first for complete documentation.

ARC.ARC - All ARC files. This should give you the basics to ARC and un-ARC anything.

ARCSHELL.ARC - This is a GEM based shell program for ARC.

GEM4ARC3.ARC - This is a GEM menu for ARC.TTP.

GEM4ARCX.ARC - A GEM interface for ARCX.TTP, the extractor program.

NOVERIFY.PRG - Turns off write verify for faster disk access.

SCRSAVER.ARC - An accessory to turn off the monitor without shutting down the system.

FX.ARC - More printer utilities for Epson FX-80 or Star NX-10 than a person can use in a lifetime.

VOLUME.ARC - Changes the name on disks without reformatting them. Great to use with the FAST-FORMAT Public Domain program from the JACG library.

SECTORED.ARC - Basically a simple sector editor. If you are familiar with these type of programs, easy to use. If you are like me, it becomes one more program to try and understand. I am real good at trashing sectors now.

UNDELETE.ARC - My kind of program. Helps me recover what I thought I no longer needed. It does work most of the time.

DEGASAVE.ARC - Saves screen to DEGAS format. This way I can save all my mistakes and view them at my leisure.

HIGHDUMP.ARC - Prints those High-rez screens in High resolution. It only uses one line of pins at a time, so be sure and rotate your pins more often than your tires. Makes great gray and white prints of my ST Writer screens.

DSLIDE.ARC - View Neo, DEGAS or Tiny pictures with titles in any resolution. I can use my beloved SM 124 monochrome monitor to view any pictures, and if I squint, they are ALL in color. Nice program.

DISKFIX.ARC - Several utilities to help you keep an eye on your disk drive. They tell about little problems before they become big ones. But they didn't fix my disk drive. I had to take it to the vet for that operation."

DOS XE

Neil Van Oost Jr. - JACG

I just received a copy of Atari's new DOS XE along with its 135 page manual. DOS XE is a renamed version of A-DOS which Bill Wilkinson of Optimized Systems Software was working on for Atari.

The new DOS XE has some good points and some bad points. First the bad points. It will not load and run on the older 800's or 400's nor is it compatible with DOS 2.0 or DOS 2.5. Although loading a special file will allow access to DOS 2.x formatted disks, the reverse is not possible. Atari used a new and different format scheme, which makes DOS XE totally incompatible with hundreds of thousands of loyal 800 and 400 ownwers machines.

On the good side, owners of the XF551 disk drive now have an Atari DOS which will support their drive. It will allow writing a true double density, using both sides of the disk for a full 360K. You will also be able to configure the DOS to read single density 707 sector disks, density and a half 1050 density disks and double density US Doubler 1050 enhanced disks.

The menu system is in four sections. The first, or main menu, gives access to the other three. They are the File access menu, the Machine language Access menu and the System function menu. The DOS is RAM resident so accessing it from BASIC is as simple as typing DOS and hitting the return key. The File access menu allows file listings to any device you specify, protecting and unprotecting of files, erase and rename of files, copy files, append to a file, initialize a disk, view the contents of a file and three directory functions. You can create a new sub-directory, change directories or delete directories.

The next menu, Machine language access menu, allows running or just loading a binary file, saving memory or appending memory to a binary file, displaying or changing memory and going to a machine language program. It also allows files listing or changing the working directory.

The last menu, System function menu, allows you to run a batch file, set the current date, initialize a disk, create a DOSXE.SYS file, duplicate a disk and allow DOS 2.x access. It also, as do the other two menues, allows file listing and changing the working directory.

The manual is fairly comprehensive and well written. It should be fairly easy for a novice to read, but MUST BE READ in order to fully understand DOS XE's workings. My first few attempts at using the DOS without reading the manual resulted in pure frustration. A lot of options which were fairly easy to perform in the DOS 2.x's have been changed. If you are an occasional Sparta Dos user such as myself, you will need to first move your Sparta files to DOS 2.x format before you can move them to DOS XE format.

All in all, my first impressions with DOS XE were negative, probably because I use my 800 as much as my 130XE and am constantly switching disks between them. Give me my DOS 2.x, Smart DOS, Super DOS, MYDOS and Sparta DOS any day. I would not be suprised if DOS XE goes the way of DOS 3.0 and the DO DO.

ACTION! FOR BASIC PROGRAMMERS

Part 3: ACTION! Statements

Dave Arlington - JACG

I'm really pleased with this month's column because I feel it has a little bit of everything for everybody who is interested in programming, whether it is in BASIC or in ACTION! For the main audience, people who want to learn ACTION!, we discuss some of the ACTION! statements and how we put them together to make programs. For TURBO BASIC people, we have information on using some of TB's special features. For ATARI BASIC fans (and I hope I'll convert you yet!), we show how to duplicate some of Turbo Basic's nice features. Finally, for all programmers, we discuss sorting techniques.

What happened was that I wanted to write a full length program in ACTION! this month for the column to show how statements in ACTION! are put together to make programs. In the meantime, I was assigned the project of

writing a sort in machine language for my college class. I figured if I could write it in assembly, I should be able to write it in ACTION! So I did. While I was at it, I wrote versions of the same sorting technique in TURBO BASIC and ATARI BASIC. I must admit that writing a program in ATARI BASIC this month for this column was a painful experience. After using sophisticated langauges like ACTION! and TURBO BASIC, using ATARI BASIC with it's all capital letters and no indenting, I felt like a primitive caveman chipping out pictures with a chisel on a stone wall. I have to say it again, get rid of your ATARI BASIC and learn TURBO BASIC at least if you still insist on using BASIC.

The other thing that occured to me as I was writing the sorting program in all three languages was that I had not seen an article on sorting techniques in a computer magazine or at a user group meeting in ages. Sure, there have been plenty of sorting programs, but little about how they work or why different sorting techniques exist. Since sorting data is something a programmer is often called upon to do, I thought I would take some time and discuss some different sorting techniques. Sorts are very useful for many things. Being able to sort numerical data in ascending or descending order or sort names or addresses in a data base are nice things to have in your programming repertoire.

But first, let's learn this month's computer buzzword. This month the magic word is 'algorithm'. Algorithm simply means a plan for solving a certain problem. For instance, suppose your neighbor's dog kept waking you up at 3AM with its loud barking. A simple algorithm to solve this problem might look like this: (written in a semi-ACTION! language)

```
Ex:1
PROC QuietDog()

  Phone(Police)
  Phone(ASPCA)
  IF NOT Police AND NOT ASPCA THEN
    Threaten(Neighbor)
  IF Neighbor=Mad OR
    NeighborSpeech="Go to hell!" THEN
      Shoot(Dog)

RETURN(Home)
Ex:2
```

By now, you're saying, "But Dave, my neighbor doesn't even have a dog, and besides I like dogs!" The point is this: Before you sit down and start writing programs at your computer, you should have some definate plan in mind on how you want to achieve your goal. For this month, the problem is to take 10 numbers from a user of your program and sort them into ascending numerical order (from lowest to highest). Let's think about some of the ways we might do this.

To get a clear idea of how your computer sorts information, imagine you are sitting at a table with a bunch of large wooden blocks with different numbers on them. The blocks are all in a straight line but the numbers are all jumbled up so they are not in any order. How would you sort the blocks so they line up in the right order?

Well, the first thing you might do if your table was big enough would be to start a new line of blocks by going through the line picking out the smallest block each time and putting it into a new line. When you get done moving all the blocks, you'll have a line with all the blocks in the right order. The trouble with this method is twofold. First, you have to have a pretty large table so you have room for two lines of blocks. Secondly, if you have 50 blocks, then you have to look through the first line 50 times to get the smallest one each time.

Another method might be to look for the smallest block in the line again, but this time, instead of moving it into a new line, swap it with the block that is already in the first position. Now you know the smallest block is in the right place. The next time through the line you start with the second block and look for the next smallest block to swap it with. Each time you move up the line one place and look for the next smallest block to put in that place in the line. By the time you get to the end, the line will be all sorted. This method is a little better since we don't waste as much space on the table and also becuase we look through a line that is one block shorter each time we go through it.

Let's improve that last method just a little bit more. This time we are going to do it a little different. We are going to look for the highest block and put it at the end of the line instead of looking for the smallest and putting it at the beginning. So we start at the same end we did before and start looking at the numbers on the blocks. Now as you are looking through them, you notice that two blocks sitting next to each other are out of order. For instance, [45] [21]. Hmmm.... We are going to have to switch those two sooner or later, so why not do it now? Now we'll have [21] [45]. What's next to the [45] block now? If it's something smaller, we'll swap them again. Now here's the beauty of this method. If we go through the line of blocks and swap blocks each time the one next to it is smaller, then we don't have to even remember what the largest block is!! It will automatically 'bubble' up to the top of the line every time! Don't believe me? Try it yourself on the blocks below:
[32] [17] [44] [3] [22] [10]

OK, we swap the 17 and 32 since 17 is smaller than 32. We leave the 32 and 44 alone since 44 is not smaller than 32. Now we swap the 3 and the 44 becuase the 3 is smaller. And so on until we end swapping the 44 and the 10 and voila, the 44 (the largest value) is now where it is

supposed to be. Works every time. Now we do it again only we only go up to block second from the end since we know the 44 is always going to be in the right place.

This method has two advantages over the previous one. First, since we are swapping blocks as we look for the largest, we are doing some preliminary sorting of the blocks that we did not do in the other two methods. Secondly we do not have to go through the line as many times as we do with the other two methods. Why not? Well, because we keep track of whether we swapped any blocks or not. If we go through the line of blocks and we don't have to swap any, that can only be because they are all in the right order! We're done! And we might not have to go through the list as many times as in the other two types of sorts.

Now turn that line of blocks into an array of numbers and the table into your computer's memory and you might start to get the idea. By the way, all these sorts have been around a long time and they even all have names. The first one is called the Insertion Sort since we insert the values from one array into another. We don't like this one since it wastes memory (you need two arrays) and it is slow (has to go through the array as many times as you have array positions). The second sort is called the Exchange Sort (I think!) because we are exchanging values. We like this a little better since it doesn't waste as much memory, but it is still not great since it is as slow as the other one. The last and probably the most famous sort is called the Bubble Sort since the largest value 'bubbles' up to the top. This is better than the other two since it does not waste memory and it usually does not have to go through as many passes through the array.

The Bubble Sort is not the best sort ever by any means. It is very popular becuase it is easy to understand and write in a program. Other sorts include the Shaker sort, Merge sort, and my favorite, the Quick sort. Usually the more sophisticated and faster the sort is, the more difficult it is to write into a program. Some sorts work better depending on what kind of information it is sorting. For instance if the arrays will be small, then you do not need a real sophisticated sort since the time difference will be minimal. If the array is very large, another more involved sort might be called for. For instance, to go back to the blocks analogy for a moment, consider if you had a few thousand blocks on the table. Chances are you would break them up into smaller groups and sort those first rather than trying to sort them as just one big humungous pile.

I won't be going into anything more sophisticated than the Bubble sort since the intent is to teach ACTION! and not dazzle you with my programming brilliance or lack thereof. I hope this gives some of you a little more understanding on how to plan out your strategies before leaping in. If interest warrants, perhaps we will discuss some of the other sorts at a later date. For now, here is the Bubble Sort in ACTION!, TURBO BASIC, and ATARI BASIC.

Ex:1

LISTING ONE: ACTION! Bubble Sort

```
PROC BubbleSort()

BYTE Pass, Exch, Index, End,
     Left_Margin=82

INT Temp

INT ARRAY Arr(10)

Left_Margin=0 Pass=0 End=9
FOR Index=0 TO 9

    DO
      Print("Enter number from -999 to 999 -> ")
      Arr(Index)=InputI()
      PrintE("")
    OD
  DO          ; Start of the REPEAT loop
    Exch=0
    FOR Index=0 TO End
      DO
        IF Arr(Index)>Arr(Index+1) THEN
          Temp=Arr(Index+1)
          Arr(Index+1)=Arr(Index)
          Arr(Index)=Temp
          Exch==+1
        FI
      OD
    Pass==+1 End==-1
    PrintF("This was pass number %B%E",Pass)
    PrintF("%B exchanges were made.%E",Exch)
    PrintE("The array looks like:")
    FOR Index=0 TO 9
      DO
        PrintI(Arr(Index))
        Print(" ")
      OD
    PrintE("") PrintE("")
  UNTIL Exch=0  ; End of REPEAT loop
  OD
  PrintE("The sort is finished!")

  RETURN
```

LISTING TWO: TURBO BASIC Bubble Sort

```
10 EXEC BUBBLESORT
20 END
100 PROC BUBBLESORT
110   DIM ARR(9)
120   LEFT_MARGIN=82
130   POKE LEFT_MARGIN,0:PASS=0:LET FINISH=9
```

```
140   FOR INDEX=0 TO 9
150     INPUT "Enter number from -999 to 999 ->";TEMP:?
160     ARR(INDEX)=TEMP
170   NEXT INDEX
180   REPEAT
190     EXCH=0
200     FOR INDEX=0 TO FINISH-1
210       IF ARR(INDEX)>ARR(INDEX+1)
220         TEMP=ARR(INDEX+1)
230         ARR(INDEX+1)=ARR(INDEX)
240         ARR(INDEX)=TEMP
250         EXCH=EXCH+1

260       ENDIF
270     NEXT INDEX
280     PASS=PASS+1:LET FINISH=FINISH-1
290     ? "This was pass number ";:? PASS
300     ? EXCH;:? " exchanges were made."
310     ? "The array looks like:"
320     FOR INDEX=0 TO 9
330       PRINT ARR(INDEX);:PRINT " ";
340     NEXT INDEX
350     ? :?
360   UNTIL EXCH=0
370   PRINT "The sort is finished!"
380 ENDPROC
```

LISTING THREE: ATARI BASIC Bubble Sort

```
10 GOSUB 110
20 END
100 REM **BUBBLESORT**
110 DIM ARR(9)
120 LEFTMARGIN=82
130 POKE LEFTMARGIN,0:PASS=0:FINISH=9
140 FOR INDEX=0 TO 9
150 ? "Enter number from -999 to 999 ->";
160 INPUT TEMP:ARR(INDEX)=TEMP:?
170 NEXT INDEX
180 REM **REPEAT LOOP**
190 EXCH=0
200 FOR INDEX=0 TO FINISH-1
210 IF ARR(INDEX)<ARR(INDEX+1) THEN GOTO 270
220 TEMP=ARR(INDEX+1)
230 ARR(INDEX+1)=ARR(INDEX)
240 ARR(INDEX)=TEMP
250 EXCH=EXCH+1
270 NEXT INDEX
280 PASS=PASS+1:FINISH=FINISH-1
290 ? "This was pass number ";:? PASS
300 ? EXCH;:? " exchanges were made."
310 ? "The array looks like:"
320 FOR INDEX=0 TO 9
330 PRINT ARR(INDEX);:? " ";
340 NEXT INDEX
350 ? :?
360 IF EXCH>0 THEN GOTO 190
370 PRINT "The sort is finished!"
380 RETURN
```

Ex:2

Let's jump right into the supposed subject for this month; ACTION! statements. We'll start by looking at something you are familiar with, Atari BASIC. BASIC has about 84 words in its vocabulary that are called reserved words or key words. These are the words you use to write a program in BASIC. No more, no less. If you take a BASIC program and strip away all the variables and constants, those words will be what you have left. For instance, in listing 3 above, some of them are GOSUB, DIM, INPUT, POKE, FOR-NEXT, IF-THEN, END, REM, ?, PRINT, =, <, and arghhh, GOTO.

One of the reasons I never liked BASIC was its limited vocabulary. If you ask someone to write a composition on what they did over their summer vacation, you'd probably get some nice reading. Tell the same people to rewrite their compositions using only 84 words that YOU give them off a list and they'd probably have a pretty tough time. Now they'd spend more time looking for the right way to put the limited vocabulary together to get their ideas across than they would actually thinking about what they were going to write!

ACTION! gets around that problem. It gives you a very small vocabulary to start with. (Next month we'll have the complete list) The big difference is that you get to put those words together to make up brand new words that you can use. Next month we'll start doing that.

There are three types of reserved words in ACTION!. First are those that are a part of the language like the key words in BASIC. These will always be printed in our ACTION! listings as all capital letters. For instance, in listing 1, the reserved ACTION! words are PROC, BYTE, INT, DO, OD, IF, FI, ARRAY, FOR, =, <, ==,TO, UNTIL, and RETURN. The second type are called library routines. These are words written by the authors of ACTION! to help make your life easier. They are built into the cartridge even though they are not an actual part of the language. You can use them just as if they were part of the ACTION! list of reserved words. They are shown with the first letter capital and the rest of the letters lower-case, followed by a parentheses that might or might not have something in it. Looking again at Listing 1, some of these are PrintE(), PrintF(), Print(), InputI().

The third type of word in ACTION! is those words we write ourselves. These can be used in other programs if written properly and next month we'll do just that. They are shown just like the ACTION! library routines.

Two last details this month. First, notice I almost snuck a new variable we haven't talked about yet into our program. We can declare arrays right now of any of the varaible types we have learned so far. That is you can declare ARRAYS of BYTES, or ARRAYS of CARDS, or in this

program, and ARRAY of INTs. Unlike ATARI BASIC, it is perfectly OK not to say what size your Array will be. For instance,

BYTE ARRAY channels

would declare an array of BYTEs without saying how many places can be in this array. You will see why this can be very useful a couple months from now. You CAN tell ACTION! how big you expect the array to be by doing just as we do in our program above, by putting the expected size in parentheses. CARD ARRAY scores(35) would declare an array of CARDs that could hold 35 CARD values (See last month's column for an explanation of BYTEs, CARDs, and INTs.)

The last thing we can do with an array is initialize all the values in the array when we declare it by putting the initial values in brackets just like we do with single variables. For example: BYTE ARRAY nums[0 0 3 4 5] would declare and array of BYTEs that is 5 long and contains the following values: nums(0) would be 0, nums(1) would also be zero, nums(2) would be 3, nums(3) would be 4, and nums(4) would be 5. Notice that like ATARI BASIC, arrays start at zero, not one. So in our example program Arr is ten spaces long, but it runs from 0 to 9 and not 1 to 10.

Last but not least, if that bubble sort written in ATARI BASIC does not make you want to trash your ATARI BASIC cartridge, I don't know what will. First off, it is nowhere near as readable since it does not indent loops and forces you to use all capital letters. The line numbers mean if you want to use it in another program, you have to make sure none of the line numbers conflict with the other program. Since it does not include such niceties as REPEAT-UNTIL or IF-ENDIF or IF_THEN_ELSE_ENDIF structures, you have to resort to contorting your program with GOTO statements and wierd conditionals. I think anyone who knew nothing about programming would find the first two listings pretty easy to follow and the last one impossible. And computer companies wonder why more people don't program when they insist on giving them BASIC to learn with. It's like going up to bat with two strikes already on you and you're the visiting team in the MetroDome!

Well, stay tuned for next month as we go deeper into the the different ACTION! statements and how they are used. Also, more BASIC-bashing! See ya!

THE COMPUTERS HAVE TAKEN OVER

by Donald Forbes - JACG

One day, we all agree, the computers will take control of our lives. The news is: That day has already arrived.

How can one tell? Well, as we all know, science is prediction, and prediction is control. And, or course, the language of science is mathematics.

The principal output of the applied mathematician is the solution of differential equations: equations involving rates of change of many variables (also known as "partial differential equations").

Now they are being solved by computer, because the work is too difficult and time-consuming and error-prone to be done by hand. So, in that sense, the computers have taken over the most important task of mathematics (the language of science) and, what is more important, there is no turning back.

You may have suspected as much all along, but the proof is finally at hand. You will find it in the September 1988 issue of SIAM Review (a journal of the 8,000-member Society for Industrial and Applied Mathematics, with headquarters in Philadelphia) in an article by Fritz Schwarz of the Society for Mathematics and Computers in West Germany. The title of the article is "Symmetries of Differential Equations: From Sophus Lie to Computer Algebra" in which he argues (with appropriate professional restraint) that "the application of computer algebra systems has qualitatively changed this area of applied mathematics."

Schwarz wrote the article after spending six months as a Visiting Scientist at the Computer Algebra Group of Dr. R. D. Jenks at IBM's Thomas J. Watson Research Center in Yorktown Heights, NY.

"During the last two decades," he writes, "a change has occurred in applied mathematics that is probably even more severe than the introduction of computers for performing numerical calculations about forty years ago. That means that large computers have rendered it feasible to perform analytical calculations as well. Although the idea of mechanizing analytical calculations is already more than 100 years old -- Charles Babbage was apparently the first person who carried it out by constructing his so-called analytical engine -- it took about 150 years after that until the large computer algebra systems for electronic computers came to be at our disposal.

"The most important general-purpose computer algebra systems today are MACSYMA at MIT, REDUCE at the Rand Corporation, MAPLE at Waterloo, mu-Math at the Soft Warehouse in Honolulu, SMP by Steve Wolfram, and SCRATCHPAD at IBM..."

"The availability of these computer algebra systems has a particularly strong influence on those areas of applied mathematics where large analytical manipulations are necessary for obtaining a certain result. Applying

the computer algebra system means to become accustomed to a completely new working style. Pencil and paper work is almost completely eliminated. Instead of working out a problem in the old-fashioned way over and over again with varying input, a solution strategy by applying computer algebra methods is developed...

"Typically this is done interactively on a computer terminal. A single procedure call may easily be equivalent to dozens or even hundreds of pages of pencil and paper work, but without any error...It makes it possible to efficiently try alternatives which take into account the results of previous runs...In this sense a keyboard and a screen may literally substitute for pencil and paper...

"It is the purpose of this article to demonstrate this new working style for the symmetry analysis of differential equations...It is general enough to be of interest to any person working with differential equations. Second, the number of calculations necessary to determine the symmetries of a differential equation is very extensive in all nontrivial cases. The computer algebra packages which have been developed for this purpose perform almost all necessary calculations completely automatically and have been extensively tested in many applications.

"The concept of the symmetry of a differential equation was introduced by (Marius) Sophus Lie at the end of the last century while he was searching for a general theory of solving differential equations. The reason his theory did not receive the proper attention for a long time is quickly realized if we try to apply it to specific problems. To find the symmetry group of a differential equation almost always requires tremendous algebraic calculations. Especially in partial differential equations, they often assume such proportions that they cannot be performed in the conventional way. The largest number of algebraic calculations is required for solving a system of linear partial differential equations which may have a simple structure; however, it may comprise several dozens or even hundreds of equations. The solution algorithm for this so-called determining system is the heart of the symmetry packages. In addition to providing the symmetry generators of the full symmetry group, its structure is determined automatically and communicated to the user in terms of its commutator table...

"It appears that specific interest in the work of Sophus Lie, and more general interest in all aspects related to symmetries of differential equations, has increased considerably in recent years...

"Solving ordinary differential equations is one of the most important problems in applied mathematics and mathematical physics since Leibniz and Newton introduced the concept of the derivative and integral of a function about 300 years ago. In the course of time there have been developed several ad hoc integration methods for special classes of ordinary differential equations which occurred in the description of physical phenomena. Naturally there arises the question of what feature distinguishes a differential equation, allowing the expression of its general solution in closed form. More generally, this amounts to asking whether there may exist a systematics of the integration of differential equations. This was essentially the state of affairs which Sophus Lie came upon when he became interested in this problem at the end of the last century. The most important recognition of Sophus Lie was that the existence of so-called symmetries plays a fundamental role in answering this question.

"This is best explained in his own words: 'Previous investigations on ordinary differential equations as may be found in customary textbooks do not form a systematic entirety. Special integration theories have been developed, e.g., for homogeneous differential equations, for linear differential equations and other special forms of integrable differential equations. The mathematicians failed to observe, however, that the special theories may be subordinated to a general method. The foundation of this method is the concept of an INFINITESIMAL TRANSFORMATION and closely related to it the concept of a ONE-PARAMETER GROUP.'

"Therewith," Schwarz continues, "for the first time a GENERAL PRINCIPLE was found. According to it the solution of a differential equation may be found SYSTEMATICALLY, if it exists at all in closed form in terms of known functions. This was a tremendous step forward, and Lie became one of the greatest mathematicians in history for this discovery, which was accompanied by his developing his theory of continuous groups... For a long period of time Lie's work was almost forgotten. Many textbooks on differential equations do not even mention Lie's concept of a symmetry at all. This is even more amazing because the knowledge of the symmetry gives a fundamental insight into the structure of the solution set and one might say that a process and a differential equation which describes it is well understood only after all symmetries which are allowed by it have been found."

As examples Schwarz points out that if the solution presents (1) a series of parallel lines in the plane, the lines can be moved up or down, or left and right without changing the essential nature of the problem, or (2) if the lines radiate as spokes from the center of a wheel, they can be expanded or contracted without a fundamental change.

"The symmetry analysis...as described...is a typical

example of how a subfield of mathematics may change in response to a computer algebra system.

"A second consequence of the availability of a computer algebra system is that the user can afford to do computer experiments without serious consideration as to whether each case is really worth the effort. In numerical computer applications it is exactly this possibility that leads to fundamentally new knowledge. As an illustration consider the numerical computer experiments of Henon, about 20 years ago, which were the foundation of nonlinear dynamics, or the discovery by Feigenbaum of the fantastic properties of nonlinear recursions.

"Both inventions certainly would not have been made without a computer because nobody could have afforded the expense for these 'useless' experiments. These are typical examples of how a purely quantitative change may finally change into a qualitative one. The applications of computer algebra systems have reached this threshold today..."

Fritz Schwarz concludes with the fascinating idea of turning the problem inside out: "The reverse problem of the symmetry analysis as described in this article consists of prescribing a symmetry group and asking for the most general differential equations which allow this invariance group. An extended (computer) package which supports this reversed problem by a suitable set of procedures will be included" in a software package which is now in preparation in West Germany.

To put the significance of Schwarz's paper in the proper pespective, it is worth taking a backward look at some of the long history of mathematics.

Italian mathematicians in the 1500's found ways to solve third- and four-degree algebraic equations (equations involving x to the third and fourth powers). Specifically, around 1515 Scipione del Ferro, professor at the University of Bologna, managed to solve the cubic equation.

The success of the Italian mathematicians produced great repercussions. Modern science now for the first time had exceeded the achievements of the ancients. Until then, in the whole course of the Middle Ages, the aim had always been only to understand the work of the ancients, and now, finally, certain questions were solved which the ancients had not succeeded in conquering. And this happened in the 1500's, i.e., in the century before the invention of new branches of mathematics: analytic geometry, differential calculus, and integral calculus, which finally affirmed the superiority of the new science over the old. After this, there was no important mathematician who did not attempt to extend the achievements of the Italians and to solve equations of fifth, sixth, and higher degree in an analogous way by means of radicals.

There was no progress for about 200 years, until Count Joseph Louis Lagrange (1736-1771), who was chiefly responsible for the establishment of the metric system, observed that the theory of permutations of roots of an equation might be of great importance in the theory of the solution of equations. He even expressed the thought that the theory of permutations of the roots of the equation is the "true philosophy of the whole question," in his book on "Reflections on the solution of algebraic equations" which was published in 1770-1771. It turned out that he was completely right, as was shown in the later investigations of Galois.

Modern algebra begins with Evariste Galois. With Galois, the character of algebra changed radically (from the theory of equations to abstractions: groups, rings, fields and algebras). Before Galois, the efforts of algebrists were mainly directed toward the solution of algebraic equations. Galois, on the other hand, was the first to investigate the structure of fields and groups, and he showed that these two structures are closely connected. If one wants to know whether an equation can be solved by radicals, one has to analyze the structure of its Galois group. After Galois, the efforts of the leading algebrists were mainly directed towards the investigation of the structure of rings, fields, algebras, and the like.

The Galois theory of equations is one of the most beautiful parts of mathematics and one of the roots of modern algebra.

Galois, who was a contemporary of Nils Abel, discovered that the heart of the problem lies in group theory, but it took 50 years before this was generally realized.

The connection between solvability by radicals and the Galois group is given by the following theorem by Galois (1830): When P is a rational prime polynomial the equation $P(x)=0$ is solvable by radicals if and only if the corresponding Galois group is solvable.

This elegant theory is the work of the tormented genius, Evariste Galois (1811-1832), whose brief life is the most tragic episode in the history of mathematics. Persecuted by stupid teachers, twice refused admission to the Ecole Polytechnique, his manuscripts rejected, or even worse, lost by the learned societies, Galois in bitterness immersed himself in the radical politics of the revolution of 1830 and was imprisoned.

Upon his release he got involved in a duel and was fatally wounded, dying before his twenty-first birthday. His manuscripts, hastily scribbled in prison and on the eve of his duel, did not receive the attention they deserved until they were read by Liouville in

1846. Only in 1962 was the critical edition of all Galois's writings finally published, but his reputation as a genius of incredible power has been secure for over a hundred years.

We must also bear in mind that there was no adequate discussion of differential equations before Cauchy in the 1820's obtained the first existence theorems. This originated the second stage. The third stage opened in the 1870's and 80's with the application by Marius Sophus Lie (1842-1899, Norwegian) of his theory of continuous groups to differential equations. Here the aim was to construct for linear differential equations an analogue of the Galois theory of algebraic equations.

It was Galois, who was Lie's idol, who indirectly inspired the application of continuous groups to differential equations. In a letter of 1874 to A. Mayer, Lie observed that "In the theory of algebraic equations before Galois only these questions were proposed: Is an equation solvable by radicals, and how is it to be solved? Since Galois, among other questions proposed is this: How is an equation to be solved by radicals in the SIMPLEST way possible? I believe the time has come to make a similar progress in differential equations."

In the late 19th century Lie considered Lie groups (then called continuous groups) for the first time. His motivation was to treat the various geometries from a group-theoretic point of view and to investigate the relationship between differential equations and the group of transformations preserving their solutions.

Despite the theoretical merits of Lie's theory, it had no value for computational purposes. Almost 100 years had to pass before it found a practical application. Eric Temple Bell in "The Development of Mathematics" (1945) pointed out at the time that: "The Lie theory of transformation groups is the implement of calculation for obtaining the covariants and invariants as necessary preliminaries to the geometry. Probably almost anyone who has ever seriously attempted to solve differential equations by the Lie theory will appreciate the labor inherent in any such heroic project...and agree with Galois that, whatever the nature of its unchallenged merits, the theory of groups does not afford a practicable method for solving equations.

"Galois of course was speaking of algebraic equations, but his opinion, in the judgment of experts in the Lie theory, carries over to differential equations. Beyond a not very advanced stage of complexity, the calculations become prohibitive to even the most persevering obstinacy."

So today the dream of Evariste Galois and Marius Sophus Lie has at last been fulfilled. The computers have taken over. And here is the historical chain: from the Ancient Greeks, to Scipione del Ferro and his solution of the cubic equation (around 1515), to Evariste Galois (1830) and the solution of algebraic equations by groups, to Charles Babbage and his "analytical engine," to Marius Sophus Lie and the solution of differential equations by continuous (Lie) groups, to IBM, and finally to Fritz Schwarz.

Interest rate is 0.12
Compounded 12 times per year
Tax Rate is 0.00
With 2000 Added Each Year For 44 Years

Year	Principal	Grows to
1	2000.00	2253.65
2	4253.65	4793.12
3	6793.12	7654.66
4	9654.66	10879.11
5	12879.11	14512.50
6	16512.50	18606.[illegible]
7	20606.70	23220.15
8	25220.15	28418.69
9	30418.69	34276.54
10	36276.54	40877.32
11	42877.32	48315.23
12	50315.23	56696.47
13	58696.47	66140.65
14	68140.65	76782.59
15	78782.59	88774.19
16	90774.19	102286.63
17	104286.63	117512.79
18	119512.79	134670.00
19	136670.00	154003.17
20	156003.17	175788.28
21	177788.28	200336.29
22	202336.29	227997.59
23	229997.59	259167.04
24	261167.04	294289.56
25	296289.56	333866.[illegible]
26	335866.49	378462.[illegible]
27	380462.77	428714.9[illegible]
28	430714.97	485340.4[illegible]
29	487340.41	549147.3[illegible]
30	551147.38	621046.6[illegible]
31	623046.66	702064.5[illegible]
32	704064.57	793357.5[illegible]
33	795357.58	896228.8[illegible]
34	898228.83	1012146.7[illegible]
35	1014146.73	1142765.9[illegible]
36	1144765.92	1289950.8[illegible]
37	1291950.89	1455802.6[illegible]
38	1457802.60	1642688.4[illegible]
39	1644688.46	1853276.1[illegible]
40	1855276.12	2090571.5[illegible]
41	2092571.57	2357962.0[illegible]
42	2359962.03	2659264.2[illegible]
43	2661264.28	2998779.2[illegible]
44	3000779.20	3381353.1[illegible]
45	3381353.12	3810193.3

Interest rate is 0.12
Compounded 12 times per year
Tax Rate is 0.25
With 2000 Added Each Year For 44 Years

Year	Principal	Grows to	Tax
1	2000.00	2253.65	63.41
2	4190.24	4721.66	132.86
3	6588.81	7424.43	208.91
4	9215.53	10384.29	292.19
5	12092.10	13625.68	383.40
6	15242.28	17175.39	483.28
7	18692.11	21062.74	592.66
8	22470.08	25319.85	712.44
9	26607.41	29981.89	843.62
10	31138.27	35087.38	987.28
11	36100.10	40678.50	1144.60
12	41533.90	46801.44	1316.88
13	47484.56	53506.79	1505.56
14	54001.23	60849.94	1712.18
15	61137.76	68891.56	1938.45
16	68953.11	77698.09	2186.24
17	77511.84	87342.28	2457.61
18	86884.67	97903.82	2754.79
19	97149.04	109469.97	3080.23
20	108389.73	122136.27	3436.63
21	120699.63	136007.37	3826.93
22	134180.43	151197.87	4254.36
23	148943.51	167833.28	4722.44
24	165110.84	186051.02	5235.05
25	182815.98	206001.62	5796.41
26	202205.21	227849.89	6411.17
27	223438.72	251776.34	7084.41
28	246691.93	277978.65	7821.68
29	272156.97	306673.28	8629.08
30	300044.20	338097.32	9513.28
31	330584.04	372510.37	10481.58
32	364028.79	410196.75	11541.99
33	400654.76	451467.81	12703.26
34	440764.55	496664.53	13974.99
35	484689.53	546160.30	15367.69
36	532792.61	600364.05	16892.86
37	585471.19	659723.59	18563.10
38	643160.49	724729.33	20392.21
39	706337.12	795918.35	22395.31
40	775523.04	873878.78	24588.93
41	851289.84	959254.70	26991.21
42	934263.49	1052751.48	29622.00
43	1025129.48	1155141.56	32503.02
44	1124638.54	1267270.86	35658.08
45	1231612.78	1387812.11	39049.83

ZMAGAZINE'S

Z*NET

THE INTERNATIONAL ATARI NEWSLETTER

March 1989 - Vol.I No.1

This Issue:

WHAT IS Z*NET?

by Ron Kovacs

Z*Net is NOT:

A) Slavic for "just say no" (znyet).

B) A fruit-fly (gnat).

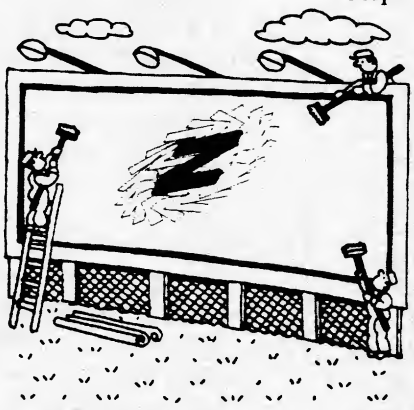
C) An epidermal eruption (zit).

Z*Net is:

A) An Interational news supplement for Atari User Groups worldwide.

B) An open communication forum.

C) Brokered advertising to support Atari User Group newsletters.



Z*Net is a totally new concept in ATARI news distribution. The Z*Net idea was born after years of discussion and user group inquiries about reprinting information from ZMagazine, my weekly online publication devoted to Atari 8 Bit news and reviews. Along with ZMag is STZMag, which covers the ST side of Atari. These publications are available on the pay telecommunication services such as CompuServe and GEnie, and over 300 bulletin board systems around the world. As we start our 4th year of online publishing, we premiere Z*Net.

Z*Net provides what many newsletter editors have been crying for: *material!* We have current news and articles from respected writers in the Atari community. Already committed to write original material for Z*Net are Ron Luks, Darek Mihocka, Darlah Pine, Matthew Ratcliff, Alan Reeve, David Small, and many others including our own staff. Of course, we encourage contributions from our readers and participating groups. We also expect to maintain direct input from Atari Corp.

The advertising money generated by this insert actually goes to the user groups that include Z*Net in their newsletter! The combined buying potential of thousands of user group members makes Z*Net ad sales much easier than individual efforts, and much easier on both the groups and the advertisers.

If you are one of the fortunate Atari Club members reading this introductory editorial, you are a member of one of our limited debut groups. As we grow and develop, we will bring you the best publishing yet offered to the *ATARI* community. If you are an advertiser or are reading this through a newsletter exchange and want more information, please contact John Nagy at (517) 487-5646, or send email via CompuServe 71777,2140, or GEnie at address ZMAGAZINE. We expect to add dozens, maybe hundreds of user groups to the Z*Net family in the coming year. See John's User Group column inside this issue for more details.

So, sit back and enjoy reading our first Z*Net. And PLEASE comment! Also debuting shortly will be our new Z*Net area on GEnie. Next month we will give you more details on this area, and provide you with information on signing up to all of the telecommunication services.

• Z*Net is a *not for profit* monthly newsletter supplement, produced by Syndicate Publishing Company. It is available only in newsletters of registered Atari user groups. Z*Net and SPC are not affiliated with Atari Corporation.

• Z*Net is produced on an Atari Mega ST4 and uses the Timeworks Desktop Publisher DTP software. It has been printed on the Hewlett Packard DeskJet printer and reproduced by user groups. Art and logos have been produced on the Atari ST using Easy – Draw and Touch – Up by *MIGRAPH.*

Z*Net
Syndicate Publishing Co.
P.O. Box 74
Middlesex, NJ 08846 – 0074

EDITOR.....Ron Kovacs
PRODUCTION Bruce Hansford
ADVERTISING .....John Nagy

CompuServe 71777,2140
GEnie ZMAGAZINE
Source BDG793

J A C G

JERSEY ATARI COMPUTER GROUP

C/O GORSKI
P.O. Box 5206
NEWARK, NJ, 07105-0206

BULK RATE
U.S. POSTAGE
PAID

RUTHERFORD, NJ
PERMIT # 44

0205 09/30/89 3RD CLASS

[redacted]

JACG NEWSLETTER VOL. 8, NUM. 12

FEBRUARY 1989

J.A.C.G. MEMBERSHIP APPLICATION

DUES: U.S. 3rd Class Mailing, Canada, Mexico $25.00
U.S. 1st Class Mailing, Foreign subscriptions $1.00

____ Renew ____ New ____ Former
____ 8-Bit ____ 16-Bit

NAME _____
Address _____
City _____
State/ Country/ Zipcode _____
Home Phone Number _____
Date _____ 1st Class _____ 3rd Class _____

MAIL TO: JACG C/O MIKE HOCHMAN
225 MADISON ST., LYNDHURST, NJ, 07071

SUPPORT YOUR CLUB AND GET SOMETHING OF VALUE AT THE SAME TIME